**🔹 Core Functionalities**

These are the basic features to build a solid recommendation chatbot.

1. **Chat-based Interface**
   * Natural language interface using a chatbot (web or app UI).
   * Accepts user queries like “Suggest a song for working out” or “I want something calm.”
2. **User Preference Input**
   * Ask about mood, genre, language, tempo, or artist.
   * Create a user profile (name, age, music taste, favorite genres/artists).
3. **Music Recommendation Engine**
   * Based on:
     + Mood (happy, sad, romantic, etc.)
     + Genre (rock, pop, classical)
     + Tempo (fast/slow)
     + Similar artists or tracks
     + Listening history (if login-based)

**🔹 Advanced Functionalities (optional)**

1. **User Account Management**
   * User login, save preferences, history, favorites.
2. **Feedback Loop**
   * “Like/Dislike” buttons for each suggestion to improve future recommendations.

**🔹 Generative AI Functionalities**

1. **Conversational Song Discovery**

* Use **OpenAI GPT or similar LLMs** to understand context:
  + “Play something like *Tum Hi Ho* but in English”
  + “Recommend a lo-fi track to help me sleep, but nothing too sad.”

1. **Emotion & Context-Based Recommendations**

* Use LLM + sentiment analysis:
  + “I had a bad day.” → Suggest soothing tracks
  + “I’m going on a long drive.” → Suggest road trip playlist

1. **Generate Song Descriptions/Reviews**

* Generate a short description or emotion analysis of the song.

1. **Generate Custom Playlists**

* "Make a 5-song playlist for coding in Python."

**🔹 Other Features**

1. **Spotify/YouTube API Integration**

* Fetch real-time song info, charts, user playlists, and playback.

1. **Daily Song/Playlist Suggestion**

* Push a “Song of the day” or “Mood booster playlist” daily.

Stack to be used :

**Framework:**

* **Next.js:** For building the full-stack application, handling both the frontend (React) and backend logic (API Routes).
* **Frontend (Chat Interface):**

**React:** The core library used within Next.js for building the user interface.

* **Backend Logic (within Next.js API Routes):**

**Node.js Runtime:** The environment where Next.js API routes execute.

**Multi-Agent Logic Implementation:** Custom JavaScript/TypeScript functions and modules within your Next.js project to represent and orchestrate the Chat Analysis, Mood Recognition, Genre Decision, and Song Search "agents."

* **Authentication:**

**Google Authentication:** The chosen identity provider for user login.

**Next-Auth (Auth.js):** A robust library for handling authentication flows in Next.js, specifically configured with the Google Provider.

* **Data Storage:**

**Firebase:** The overall platform.

**Firestore:** A NoSQL cloud database for storing user data such as profiles, preferences, chat history, favorite songs, and feedback ("Like/Dislike").

**Firebase Admin SDK:** Used in Next.js API routes for secure server-side interaction with Firestore.

* **Generative AI: Large Language Model (LLM) Provider:** **Google AI** (for Gemini models).
* **External APIs: YouTube Data API:** For searching and retrieving information about songs and videos based on the recommendations from your AI agents.